Ref.5

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（A）　(11)特許出願公開番号

特開平8－255264

(43)公開日　平成８年(1996)10月１日

| (51)Int.Cl.[6] | 識別記号 | 庁内整理番号 | ＦＩ | | 技術表示箇所 |
|---|---|---|---|---|---|
| Ｇ０６Ｔ　15/50 | | 9365－5H | Ｇ０６Ｆ　15/72 | ４６５ | |
| 17/00 | | | 15/62 | ３５０Ａ | |

審査請求　未請求　請求項の数６　ＯＬ　（全 17 頁）

(21)出願番号　特願平8－14192

(22)出願日　平成８年(1996)１月30日

(31)優先権主張番号　９５０１８３２：１

(32)優先日　1995年１月31日

(33)優先権主張国　イギリス（ＧＢ）

(71)出願人　594128728
ビデオロジック　リミテッド
イギリス　ハートフォードシャー　ダブリューディー４　８エルゼット　キングス　ラングリー　ホーム　パーク　インダストリアル　エステイト　ユニット　８

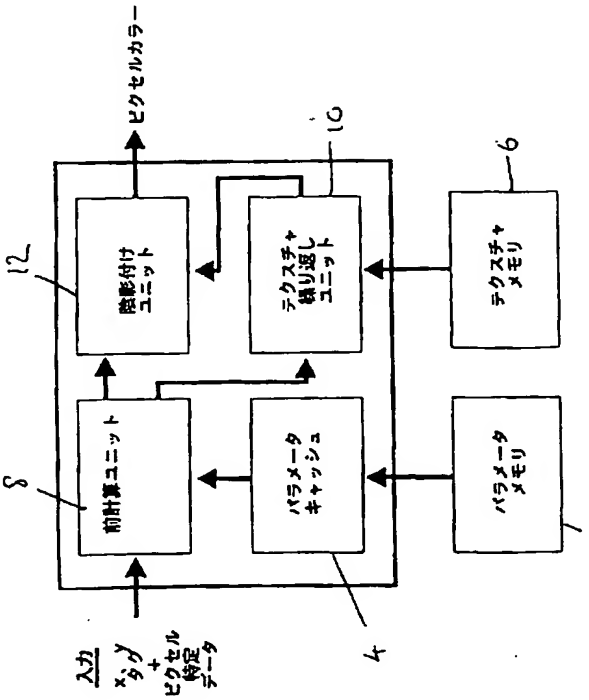
(72)発明者　シモン　ジェームズ　フェニー
イギリス　ハートフォードシャー　エスジー13　８キューアール　ストリート　ヴィレッジ　ニューゲート　ボンズボーン　マナー　５

(74)代理人　弁理士　中村　稔　（外６名）

最終頁に続く

(54)【発明の名称】　３Ｄ像のテクスチャ処理及び陰影付け方法

(57)【要約】
【課題】　スクリーンに表示するように三次元像をテクスチャ処理及び陰影付けする方法を提供する。
【解決手段】　三次元像は、スクリーンの各画素領域（ピクセル）の位置と、そのピクセルに関連した像データとを含むデータを最初に受け取ることによりスクリーン上に表示するようにテクスチャ処理される。テクスチャ像データは、像データと、そのテクスチャ像データがピクセル上にマップされる適切な位置とに基づいてメモリ手段から検索される。同じ関連像データを共有するピクセルの数が決定され、そして１つのピクセル増分に対応するピクセル上のテクスチャデータのマッピングの増分変化も決定される。同じ像データを隣接ピクセルとして共有する各ピクセルの場合は、その隣接ピクセルについて既に導出されているテクスチャ像データが、テクスチャ像データの増分変化と結合されて、そのピクセルのためのテクスチャ像データが導出される。



1

【特許請求の範囲】

【請求項１】　スクリーンに表示するために三次元像をテクスチャ処理する方法において、スクリーンに対する各画素領域（ピクセル）の位置と、ピクセルに対する関連像データとを含むデータを受け取り、上記像データに基づいてメモリ手段からテクスチャ像データを検索し、上記関連像データに基づいて上記ピクセルに上記テクスチャ像データの適切な部分をマッピングし、同じ関連像データを共有するピクセルの数を決定し、１ピクセル増加に対応してピクセルにおける上記テクスチャデータのマッピングの増分変化を決定し、そして隣接ピクセルと同じ像データを共有する各ピクセルに対し、その隣接ピクセルに対して既に導出されたテクスチャ像データを、テクスチャ像データの上記増分変化と結合して、そのピクセルに対するテクスチャ像データを導出するという段階を備えたことを特徴とする方法。

【請求項２】　像をテクスチャ処理するのに使用するＭＩＰマップデータをダイナミックランダムアクセスメモリ（ＤＲＡＭ）に記憶し、テクスチャＭＩＰマップのｕ及びｖアドレスビットをインターリーブし、これにより、物理的に隣接するＭＩＰマップピクセルが同じＤＲＡＭページに入る最大の確率を有するようにする請求項１に記載の三次元像をテクスチャ処理する方法。

【請求項３】　ＭＩＰマップデータの連続する解像度レベルを２つのＤＲＡＭバンクに交互に記憶し、第１のＤＲＡＭバンクのメモリページは、所与のテクスチャに対するＭＩＰマップデータの解像度の全ての偶数レベルを含み、そして第２のＤＲＡＭバンクのメモリページは、そのテクスチャに対するＭＩＰマップデータの解像度の全ての奇数レベルを含むようにする請求項２に記載の方法。

【請求項４】　第１のＤＲＡＭバンクのメモリページにおけるＭＩＰマップデータの解像度の偶数レベルは、第２のテクスチャに対するＭＩＰマップデータの解像度の奇数レベルとインターリーブされ、そして第２のＤＲＡＭバンクにおけるＭＩＰマップデータの奇数レベルは、第２のテクスチャに対するＭＩＰマップデータの偶数レベルとインターリーブされる請求項３に記載の方法。

【請求項５】　除算演算のためのマシンによる方法において、ｎが１より大きいとすれば、０ないしＸの範囲の各ｎ番目の値の逆数を記憶し、介在するｎ－１の値の逆数間の差を記憶し、除算演算のためのオペランド及び仮数を受け取り、受け取った仮数の最上位部分に応答して記憶された逆数の１つを検索し、受け取った仮数の最下位部分に応答して１組の差を受け取り、上記逆数と１組の差を加算し、そして加算結果を乗算するという段階を備えたことを特徴とする方法。

【請求項６】　長除演算を実行する装置において、ｎが１より大きいとすれば、０ないしＸの範囲の各ｎ番目の値の逆数を記憶する手段と、介在するｎ－１の値の逆数

2

間の差を記憶する手段と、除算演算のためのオペランド及び仮数を受け取る手段と、受け取った仮数の最上位部分に応答して記憶された逆数の１つを検索する手段と、受け取った仮数の最下位部分に応答して１組の差を検索する手段と、上記逆数と１組の差を加算する手段と、受け取ったオペランドで加算結果を乗算する手段とを備えたことを特徴とする装置。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、スクリーンに表示するために３Ｄ像をテクスチャ処理及び陰影付けする方法に係る。

【０００２】

【従来の技術】テクスチャ処理されそして陰影付けされたリアルタイムの３Ｄ像を発生するために商業用システムによって最も一般的に使用される方法は、Ｚバッファシステムを使用するものである。これは、ソフトウェア又はハードウェアのいずれかで実施するための最も簡単な可視表面アルゴリズムの１つであり、シリコングラフィックス、イベンツ＆サザーランド、及びヒューレットパッカードのような会社で使用されている。

【０００３】これは、カラー値が記憶されるフレームバッファを入手できるだけでなく、各ピクセルごとにｚ値が記憶される同じ入力数のＺバッファも入力できることが必要である。多角形、通常は三角形が、任意の順序でフレームバッファにレンダリングされる。走査ー変換の間に、多角形の点がバッファに既にある点よりも視聴者から離れない場合に、新たな点のテクスチャ処理及び陰影付けされたカラーが評価されて、ｚ値が古い値に置き換えられる。予めの分類は不要であり、オブジェクトーオブジェクトの比較も不要である。

【０００４】「テクスチャマッピング」という用語は、ピクセルの２Ｄアレーをピクセルの別の２Ｄアレーに変換するプロセスを指す。この変換は全く任意であるが、ここでは、遠近マッピングを考える。遠近マッピングは、実際には、ピクセルの２Ｄアレーを取り上げ、それらを回転して３Ｄに変換し、そしてそれらを表示のためのｚーｎ平面に投射する。

【０００５】テクスチャマッピングは、コンピュータグラフィックにおいて、実物の表面細部の模擬を試みるのに使用される。遠近マッピングを用いると、画（例えば、木の表面）を３Ｄ物体（例えば、テーブル）上に配置することができる。その結果として、テーブルは見掛けが木のようになる。

【０００６】遠近マッピングの式は、次の通りである。

$u = (ax+by+c) / (px+qy+r)$

$v = (dx+ey+f) / (px+qy+r)$

ｕ及びｖは、テクスチャスペースにおける２Ｄ座標であり、ｘ及びｙは、スクリーンスペースにおける２Ｄ座標であり、そしてａ、ｂ、ｃ、ｄ、ｅ及びｆは、マッピン

3

グプロセスに使用される係数である。

【0007】サンプリング率の半分より上の周波数をもつ信号をサンプリングすると、エイリアシングを引き起こす。テクスチャマッピングは、エイリアシングを引き起こし易い再サンプリングプロセスである。エイリアシングを生じないテクスチャマッピングされた像を形成するために、テクスチャスペースの像をマッピングのサンプリング率の半分までローパスフィルタしなければならない。この状態に対する複雑さは、スクリーンスペース内の位置に基づいてサンプリング率が変化することである。

【0008】像を正しくフィルタするために、大量の処理を必要とする。数学的に簡単な演算で正しいフィルタ動作を近似する方法は多数ある。最も簡単な方法は、ＭＩＰマッピングであり、ＭＩＰは、ＭＵＬＴＩＭ　ＩＭＰＡＲＶＯの省略形である。

【0009】ＭＩＰマッピングは、メモリに記憶されたテクスチャマップをフィルタしそして半分の解像度に減少する前処理段階である。これは、得られる像のサイズが１ピクセルになるまで繰り返される（これは、テクスチャが方形で、２の累乗であると仮定する）。図１は、煉瓦のテクスチャの例を１２８ｘ１２８の解像度及びそれに関連した低いＭＩＰマップレベルで示している。

【0010】ＭＩＰマップは、像のピラミッドとして考えることができる。ＭＩＰマップは３つの変数ｕ、ｖ及びＤを介してアクセスされる。変数ｕ及びｖは、アンチエリアシングする必要のあるテクスチャスペースにおけるピクセルの座標である。変数Ｄは、フィルタされたそのピクセルがいかに必要とされるか、即ちピラミッドがいかに高いかの尺度である。値Ｄ＝１は、全解像度の像が使用されることを意味する。値Ｄ＝２は、半分の解像度の像が使用されることを意味し、等々となる。Ｄの値＊

$$I(L,S,\theta)=\begin{cases} LS(\cos\theta) & (\cos\theta)\geqq 0\text{の場合} \\ 0 & (\cos\theta)<0\text{の場合} \end{cases}$$

【0017】図２は、目又はカメラから発生され（特定のシーンピクセルを通過する）そして表面の所与の点Ｐに当たる光線を示している。又、Ｐにおいて表面に垂直な直角のベクトル及び光源からＰへの光線も示されている。

【0018】上記式によれば、Ｐにおける拡散照明の輝度（又はカラー）Ｉは、光の輝度Ｌと、表面の拡散反射率Ｓと、光線と法線との角度$\theta$のコサインとの積である。

【0019】この計算は、表面全体に対して一定と仮定する（フラットシェーディングとして知られている）か、又は曲面を模擬するように多角形の小面にわたって種々の仕方で補間することができる。例えば、ゴーラウドシェーディング（ＣｏｍｍＡＣＭ　１８（６０）、第３１１－１７ページ、１９７１年）は、多角形の小面に

4

＊が２の累乗でないときには、２つの最も近いＭＩＰマップレベルを混合したものが、良く知られたように線型補間を用いて計算される。

【0011】理想的に、Ｄは、テクスチャスペースへと変換されるときにスクリーンピクセルがカバーする面積の平方根である。不都合なことに、これは、ピクセルごとのベースで計算するには非常に経費がかかる。

【0012】Ｐ．Ｓ．ヘクバート氏（Ｃｏｍｍ　ＡＣＭ　１８（６）、１９７５年６月）はＤに対し次の近似式を用いることを示唆している。

$D^2=MAX\left(\frac{du^2}{dx}+\frac{dv^2}{dy}\right),\left(\frac{du^2}{dy}+\frac{dv^2}{dy}\right)$

【0013】この方法は、理想的なものよりは遙に簡単であるが、依然として平方根を必要とすると共に、ピクセル当たり２つの付加的なマッピングを必要とする。

【0014】陰影付け（シェーディング）は、コンピュータグラフィックに使用される用語で、特定のスクリーン位置における面のカラーを評価するプロセスの一部分を指す。面が陰影付けされる場合には、その面と光源がいかに相互作用するかをモデリングするのに、面の位置、向き及び特徴が使用される。

【0015】コンピュータで形成される像の現実感は、この照明モデルの質によって大きく左右される。不都合なことに、精度のよい照明モデルは、リアルタイムのレンダリングシステムに使用するのに高価であり過ぎる。それ故、照明計算において近似を用いて妥協を図る必要がある。

【0016】最も一般的な近似の形態は、材料の艶消し成分から反射される光を模擬する拡散又はランバートシェーディングである。これは、図２に示されており、次の式で表される。

わたる拡散カラーの線型補間である。

【0020】別の補間方法は、上記式において表面の法線を線型補間することである。この方法は、ブイ・スオング・フォンにより提案されている（Ｃｏｍｍ　ＡＣＭ　１８（６）、１９７５年６月）。不都合なことに、この計算は、比較的経費がかかる。又、ビショップ及びウエイマによって効果的な近似が提案されており（コンピュータグラフィックス、２０（４）、第１０３－６ページ、１９７５年）、これは、二次元のテイラー級数近似を使用している。これらの近似は、差の方程式を用いて効果的に評価することができる。僅かな前計算オーバーヘッドはさておき、$\cos\theta$は、ピクセル当たり２つの加算のみを犠牲にして近似することができる。

【0021】模擬される次に最も一般的な照明特徴は、光沢のある又は鏡のようなハイライトである。これら

5

は、シーンにおける光の反射である。鏡のようなハイライトは表面に反射の見掛けを与え、像の現実感を相当に改善する。これらも、ブイ・スオング・フォンにより述＊

6

＊べられた方法によって最も一般的に近似され、これは、図3に示され、次の式で表される。

$$I(L、S、\theta、n) = \begin{cases} LS(\cos\theta)^n & (\cos\theta) \geqq 0の場合 \\ 0 & (\cos\theta) < 0の場合 \end{cases}$$

【0022】図3は、目又はカメラから発生されて、特定のピクセルを通過し、所与の点Pにおいて表面に当たる光線を示している。この光線は、次いで、通常のミラー角度を用いて図示された方向に反射される。表面は完全に滑らかではないから、反射光線は、ある程度分散すると仮定され、これは表面の粗面さによって左右される。更に、図3は、明るい光線として見える光源からの光線も示している。反射方向が光線に近いほど、ハイライトは明るく見える。

【0023】上記式は、ハイライトIの輝度を、光の輝度Lと、表面の反射率Sと、反射方向と光の間の角度$\theta$と、表面の滑らかさnとの関数として示している。この関数は、反射方向が光線方向に近いほど（即ち、$\theta$が小さいほど）、ハイライトの輝度が最大に接近し、他の領域において減少することを意味する。nの値が大きいほど、減少は速くなり、ハイライトは小さくなり、ひいては、表面の見掛けは滑らかになる。最も効果的であるためには、この計算を累乗関数と称する必要がある。nの典型的な値は、1から1000の範囲である。

【0024】又、ビショップ及びウエイマ氏は、二次元テイラー級数近似（拡散シェーディング参照）を使用すると共に、累乗関数を計算するためにルックアップテーブルを使用した上記式の計算部分の提案している。

【0025】DRAMは、ダイナミックランダムアクセスメモリの頭文字である。「ページモード」DRAMの構造は、バンク、ページ及び位置に分割することができる。位置とは、アトミックアドレスメモリ素子である。多数の隣接する位置がページを作り上げ、そして多数のページがバンクを作り上げる。

【0026】装置が位置の内容を要求したときには、その位置を含むページがオープンされそしてデータがフェッチされる。次に要求されたページが同じページ内に入る場合には、データをフェッチするまでページを再オープンする必要はない。ページをオープンするのに要する時間は著しいものであるから、ページ内のランダムな位置を要求することは、ページ外のランダムな位置を要求するよりも相当に速いものとなる。「ページブレーク」という用語は、オープンしていないページ内の位置の内容をフェッチすることを指す。

【0027】多数のバンクを有するメモリ構成においては、一度に2つ以上のページをオープン状態に保持することができる。オープンしたページは異なるバンクになければならないという制約がある。

【0028】SRAMは、スタティックランダムアクセスメモリの頭文字である。SRAMの構造は、DRAMより簡単である。装置内のいかなる位置も単一のクロックサイクル内でフェッチすることができる。この簡単化の欠点は、シリコン上のメモリ素子の密度がDRAMよりも相当に低いことである。その結果、DRAMよりも相当に高価なものになる。

【0029】霧の雰囲気効果は、発生される3Dシーンに、ある次元の現実性を付加する。霧のように見えるシーンは単に有用ではなく、全ての屋外シーンは霧を必要とする。屋外シーンにおいて、遠くにある物体は、あまり刺激的な色をしていない。この効果は霧と同じであり、物体は距離と共に「灰色化」する。

【0030】霧の密度が変化しないと仮定すれば、光は、1m当たりある割合で減衰する。例えば、光が1m当たり1％で減衰する場合には、1mで光は0．99になる。2mでは、元の光の0．99＊0．99＝0．99$^2$＝0．98となる。「n」mでは、光は元の0．99$^n$となる。

【0031】Zバッファシステムに関連したテクスチャ処理及び陰影付けに伴う問題は、各シーンピクセルがそのカラーを何回も評価されることである。これは、表面がランダムに処理され、目に最も近い面が、シーンの処理中に何回も変化するからである。

【0032】

【発明の構成】本発明の1つの特徴を用いた据え置きテクスチャ処理及び陰影付けアーキテクチャは、各スクリーンピクセルごとに単一のカラーを処理するだけで計算浪費を排除する。シーン内の各面には、独特の「タグ」が関連される。ピクセルを処理するために、隠れた面の除去が行われ、そのピクセルに最も近い面のタグが計算される。このタグの値を用いて、そのピクセルを正しくテクスチャ処理及び陰影付けする命令がフェッチされる。

【0033】ここに述べる実施形態は、据え置きテクスチャ処理及び陰影付けを必要とするシステムに特に適した非常に効率的で且つ最適なアーキテクチャを提供する。ここに述べるような複雑なシステムの場合には、複雑さ（ハードウェアサイズ／シリコン面積）と機能性との間の最良のバランスを入念に達成するよう確保することが非常に重要である。これは、主たるブロック間の相互作用と、各ブロックに使用される低レベル方法の細部とを含む全システムアーキテクチャーを選択時に入念に設計しなければならないことを意味する。特に重要なものは、次の通りである。

7

１．障害を確実に最小にするデータ編成及びメモリアーキテクチャ；
２．種々の機能ブロック、メモリキャッシュ要求及びそれらの相互作用に関するリソースの区分化；
３．数値の精度についての複雑さレベル、及び各ブロックに必要とされる機能を実施するのに用いる方法に使用される近似の程度；
４．ハードウェア／シリコン予算内での充分な融通性レベルのサポート。

【００３４】ここに述べる実施形態は、上記の事柄についてバランスをとるために入念なアルゴリズム及びシュミレーション分析を使用する。

【００３５】ＭＩＰマッピングを実行するシステムの率決定段階の１つは、メモリからテクスチャピクセルをフェッチすることである。帯域巾を広く保つために、ある設計では、高速ＳＲＡＭを使用する。これは、所望の性能を与えることができるが、このような解決策のコストは、甚だしいものとなる。

【００３６】本発明の好ましい実施形態において提案されたアーキテクチャーは、ＭＩＰ－ＭＡＰデータをページモードＤＲＡＭに記憶する。最高のデータスループットを得るために、ページブレークの数を最小に減少しなければならない。ＭＩＰマップを記憶する最も簡単な方法は、Ｄレベルを隣接配置し、そしてマップ内でピクセルが走査順序で記憶されるようにすることである。ＭＩＰマップへのアクセスの性質により、この記憶方法は、ほぼアクセスごとにページをブレークする。メモリ構造は、ページブレークの数を最適なものにする。

【００３７】上記のテクスチャ処理方程式の係数は、潜在的に境界のないものである。分かり易い境界が実施されても、必要とされる値は広い範囲に及ぶ。純粋な固定小数点演算を用いて方程式を解く場合は、高精度の乗算器及び除算器が必要となる。このようなハードウェアは、大型で低速であり、従って、システムの価格及び性能に影響を及ぼす。本発明の別の特徴による混合浮動及び固定小数点方法は、２つの利点を有する。第１に、乗算器及び除算器のサイズを縮小することができ、そして第２に、全浮動小数点演算を実施する複雑さが回避される。

【００３８】テクスチャマッピング方程式は、ピクセル当たり１つの除算を必要とする。殆どのハードウェア除算アーキテクチャーは、次々に近似する方法を使用し、これは、所望の精度に対し繰り返しを必要とする。この繰り返しは、時間的に行わねばならない（多数のクロックサイクル）か、又はスペース的に行わねばならない（シリコン領域）。次々の近似を伴わずに所望の精度の逆数を評価するための本発明の更に別の特徴による新規な方法及びアーキテクチャーを以下に述べる。

【００３９】変数「Ｄ」は、テクスチャ処理されたピクセルをいかにフィルタすることが必要かの尺度である。

8

「Ｄ」に対するヘクバート氏の近似は、計算的にかなり経費がかかり、平方根を必要とすると共に、及びピクセル当たり２つの付加的なマッピングを必要とする。「Ｄ」を計算する更に簡単な新規な方法について以下に説明する。

【００４０】テクスチャ処理及び陰影付けプロセスに使用されるパラメータをフェッチする場合には、ピクセルを処理できる率に影響が及ぶ。パラメータが必要とされる順序は、完全にランダムではなく、基準の位置を利用することができる。ここに述べるアーキテクチャーはパラメータキャッシュを含み、これは、全設計の性能を著しく改善する。

【００４１】鏡のようなハイライトのサイズの輝度を制御するために、累乗関数が一般的に使用される。これは、実施に経費がかかり、大きなＲＯＭルックアップテーブルによる近似又は正確な累乗関数の明確な実施の２つが考えられる。ハイライトの方程式は近似に過ぎないので、この関数は、完全に正確な累乗関数である必要はない。

【００４２】$\cos\theta$を計算又は近似する方法が存在すると仮定すれば、累乗関数は単に次の計算となる。

$x^n$ 、但し、$0 \leqq x \leqq 1$

【００４３】又、ｎの粒度について微細な制御を行う必要はなく、若干の整数値で充分である。次の式に注目されたい。

$(1-y)^{2n} \propto (1-2y)^{2n-1}$

ｙが小さいときには、次の近似を使用し、ｘが上記範囲内のときに、ｘを累乗ｎまで上昇させる。

$x^n \fallingdotseq (1-\max(1,\ (1-x),\ 2^k))^2$

但し、ｋは、$(\log_2 n)-1$の整数部分である。値ｋは、表面特性の一部分として値ｎに効果的に置き代わる。

【００４４】２つの減算の計算は、比較的安価である。ある値に２進表示で$2^k$を乗算することは、単にｋ個の場所だけ左にシフトすることであり、最大値の計算も平凡な演算である。平方演算は、値それ自身の乗算である。

【００４５】この方法は、ハードウェア及びファームウェアの両方の実施形態に適用可能である。

【００４６】シーンを霧の状態にするときに減衰係数を計算するには、霧の濃度を、物体と観察者との間の距離の累乗まで上昇させる必要がある。この関数を近似する効率的な方法は、以下に説明する。

【００４７】本発明の種々の特徴は、特許請求の範囲に規定する。以下、添付図面を参照して本発明の好ましい実施形態を詳細に説明する。

【００４８】

【発明の実施の形態】図４は、システムを通る主なデータの流れを示す図である。システムへの入力は、本出願人の英国特許出願第９４１４８３４．３号に開示された

システムで発生できる形式の「x、y」座標、タグ（Tag）及びある任意の「ピクセル特定データ」である。入力ピクセルデータは、「走査順序」又は「タイル順序」で受け取られる。走査及びタイル順序は、図5及び6に示されている。

【0049】「タイル順序」でデータを受け取る利点は、「走査順序」の場合よりも、像のローカルピクセルが互いに接近して送信されることである。基準の位置は、パラメータキャッシュの性能を改善する。

【0050】上記したTagは、シーンにおける各々の潜在的に見える面に関連した独特の識別子である。

【0051】「ピクセル特性データ」は、システムの外部で評価されたピクセルの属性である。これらの属性は、ピクセルカラーの評価に使用される。例えば、「三次元像の陰影付け(Shading Three-Demensional Image)」と題する特許出願（出願第９４１４８３４．３号）に開示されたシステムでは、各ピクセルに、影が投じられるかどうか記述するデータが関連される。この情報は、ピクセルカラーを評価するときに加味することができる。

【0052】ピクセルTagは、パラメータメモリ2へのインデックスとして使用される。これは、各ピクセルにおいて見える面を定めるパラメータを記憶する。インデックスされた位置に記憶されるデータは、ピクセルカラーの評価に使用される（例えば、テクスチャマッピング方程式の係数）。

【0053】Tagが2つ以上のピクセル位置に現れるときには、そのTagに関連したデータが2回以上必要となる。というのは、目に見える同じ面を表示するのに多数のピクセルが使用されるからである。それ故、パラメータキャッシュ4をデータ経路に挿入することによりフェッチパラメータの性能を改善することができる。シュミレーションにおいては、2方セット連想キャッシュが最良の複雑さ／性能比を与えている。

【0054】テクスチャメモリ6は、ＭＩＰマップデータを含むＤＲＡＭの多数のバンクで構成される。入力ピクセルの「u、v」座標が評価されると、それに関連したテクセル（Texel）（テクスチャメモリの画素領域）がテクスチャメモリからフェッチされる。

【0055】テクスチャマッピング方程式は、2つの段即ち前計算段及び繰り返し段に分割される。前計算ユニット8では、パラメータがフェッチされ、次いで、「ax＋by＋c」、「dx＋ey＋f」及び「px＋qy＋r」が評価される。又、このユニットは、いかに多くの隣接ピクセルが同じTag値を共有するかをカウントする。このカウントは、上記の値と共に、テクスチャ繰り返しユニット10へ転送される。

【0056】テクスチャ繰り返しユニット10は、前計算ユニット8から値を取り上げて、2つの除算を実行し、「u、v」評価を完了する。前計算からのカウント値が0より大きい場合には、現在ピクセルと次のピクセルとの間の差（即ち、「a」、「d」及び「p」）が、前計算された値（「ax＋by＋c」、「dx＋ey＋f」及び「px＋qy＋r」）に加算され、そして2つの除算が繰り返される。この段階は、「カウント」で指定された回数だけ繰り返される。



【0057】テクスチャ繰り返しユニットからのカラーが計算されると、それが陰影付けユニット12へ通される。ピクセルTagにより指示されたパラメータは、単なるテクスチャ処理係数ではなく、それらは、陰影付けユニットにより最終ピクセルカラーを評価するのに使用する情報も含んでいる。

【0058】行われる陰影付け計算の最も簡単な形態は、一次及び二次関数陰影付けのような「x、y」スクリーン位置の関数である。

【0059】一次陰影付けは、次の式を計算する。

$$I(x,y)=T_2x+T_1y+T_0$$

但し、Iは輝度（又はカラー）であり、x及びyはスクリーンピクセル座標であり、$T_2$、$T_1$、$T_0$は定数である。これは、テクスチャ処理と同様に前計算部分及び繰り返し部分に分割することができる。

【0060】出力は、範囲0．．．1内に存在するように制約される。

【0061】二次陰影付けは、関数の良好な近似を与えるが、実施のコストを高める。これは、次の式の関数によって実施される。

$$I(x,y)=T_5x^2+T_4xy+T_3y^2+T_2x+T_1y+T_0$$

但し、Iは輝度（又はカラー）であり、x及びyはスクリーンピクセル座標であり、そして$T_5$、$T_4$、$T_3$、$T_2$、$T_1$、$T_0$は定数である。これは、テクスチャ処理と同様に前計算部分及び繰り返し部分に分割することができる。

【0062】出力は、範囲0．．．1内に存在するように制約される。

【0063】二次関数は、前記の累乗関数を用いて、ある累乗まで高めることができる。

【0064】ピクセルは、テクスチャ処理され、陰影付けされた後に、フレーム記憶装置へ出力される。次いで、フレーム記憶装置からピクセルがフェッチされて表示装置へ送られる。

【0065】本発明の好ましい実施形態である非常に最適化されたアーキテクチャが図11に示されている。

【0066】システムへの入力データ20は、ピクセルデータではなくタグをその出力として記憶する隠れた面の除去及び深さの分類装置から導出される。このようなシステムは、英国特許出願第９４１４８３４．３号に開示されている。入力データは、シーンの各面に対して独特の表面タグと、システムへ入力されるピクセルの現在ブロックのスタートアドレスであるブロックx、yアド

レスと、現在入力されているピクセルの深さである深さ（Ｚ）値と、現在ピクセルが１つ以上の陰影ライトの影にあるかどうかを指示する陰影フラグとを備えている。

【００６７】所与のブロックに対する表面タグは、＜表面タグ：スパン長さ（パラメータが同じに保たれるところのピクセルの数）＞としてコード化されてタグバッファ２２に記憶されるラン長さである。他の入力パラメータは、ピクセルごとのベースで入力ブロックバッファ２４に記憶される。

【００６８】スパンは、パラメータキャッシュ２６に対して比較される。ブロックはタイル状のフォーマットで入力され、従って、所与のシーンの垂直及び水平のコヒレンスがキャッシュにより利用できるのが望ましい。キャッシュの連想性の程度は、個々のシステムコスト／性能の兼ね合いについて残されている問題である。キャッシュがヒットする場合に、それに対応する表面パラメータリストは、前計算ユニット８へ直接通される。キャッシュがミスする場合には、パラメータリストがシステムメモリからフェッチされ、記憶されそして前計算ユニットへ送られる。キャッシュは、高いヒット率で設計されているので、システムメモリバスの帯域巾におけるその後のミスのペナルティは小さく、個別のパラメータメモリの必要性は否定される。

【００６９】前計算ユニットは、ブロックスタートアドレス（ここからスピンスタートアドレスが導出される）を現在パラメータリストと共に取り上げ、パラメータリストに指定されたようにテクスチャ／陰影付け繰り返し装置２８、３０のいずれか又は全てに対する初期値を計算する。このようにする際に、上記の走査線テクスチャアドレス及び陰影付け輝度アルゴリズムを実施する。これは、典型的に、マイクロシーケンサによって制御される乗算アキュムレータのアレーより成る。

【００７０】前計算バッファ３２は、キャッシュミスのペナルティ及び前計算オーバーヘッドをテクスチャ及び陰影付け繰り返しプロセスにオーバーラップさせ、これにより、システムスループットを増加できるようにする。

【００７１】テクスチャ繰り返し装置は、以下に述べるように、ｕ、ｖ及びＭＩＰマップの「Ｄ」計算において双曲線補間を実行する。テクスチャ繰り返し装置の出力は、テクスチャメモリ６のアドレスの対より成り、２つの適切なＭＩＰマップからのテクスチャ値がフェッチされてピクセル処理ユニット３４へ送られる。

【００７２】陰影付け繰り返し装置２８、３０の数は、陰影付け機能に必要とされる並列の程度によって左右される。例えば、２つのユニットは、滑らかに陰影付けされた面に対し全体的な照明輝度と影の光輝度とを同時に評価することができる。或いは又、これらユニットは、フォングの陰影付けされた面の拡散成分及び鏡状成分を並列に計算するのに使用できる。繰り返し装置は、以下



のアルゴリズムを実施する。陰影付けパイプラインの出力は、１組の輝度値を含み、これらは、ピクセル処理ユニットへ送られる。

【００７３】霧付与ユニット３６は、入力ブロックバッファから記憶された深さ値を取り上げ、そして以下に述べるように、霧付与の擬似指数関数を解く。これらの値は、ピクセル処理ユニットへ送られ、そこで、計算されたピクセルカラーと霧カラーとの間を補間するのに使用される。

【００７４】ピクセル処理ユニット３２は、各スクリーンピクセルの最終的なＲ、Ｇ、Ｂカラー評価を行う。基本カラー又はテクスチャカラーが入力され、そして陰影ビットにより指示された陰影付け繰り返し装置からの全ての当該光源輝度値の条件和で乗算される。ハイライトのオフセットも加算される。次いで、ピクセルは霧が付与され、ブロック累積バッファ３８へ転送される。

【００７５】ここに述べるアーキテクチャは、多数の進歩した特徴をサポートすることができる。これは、英国特許出願第９４１４８４３．３号に開示されたような互換性のある隠れた面の除去及び深さ分類システムを必要とすることに注意されたい。

【００７６】１）半透明の面：半透明のオブジェクトは、マルチパス技術を用いてレンダリングすることができる。先ず第１に、ブロックの不透明な面が処理され、そして上記のように累積バッファへ転送される。次いで、半透明の面が同様に処理されるが、全ての半透明の面は、それらのパラメータリスト又はテクスチャデータの一部分として「アルファ」成分を有し、これを用いて、現在ピクセルと、累積バッファに記憶された対応するバックグランドピクセルとの間で混合が行われる。全ての半透明ピクセルが処理されたときに、累積バッファの内容がフレームバッファから流出される。

【００７７】２）アンチエリアシング：これは、マルチパス技術を用いて達成することができる。この方法においては、各ブロックがＸ及びＹのサブピクセル増分で何回も通される。それにより得られるピクセルカラーは、ブロック累積バッファに累積される。パスが完了すると、ピクセルがパスの回数によって分割され、フレームバッファへ送られる。

【００７８】３）動きのぼけ：これは、時間的にぼけさせるべきオブジェクトを過剰サンプリングし、次いで、上記のようにブロック累積バッファを用いてサンプルを平均化することにより達成される。

【００７９】４）ソフト陰影：この効果は、マルチパス技術を用いて模擬することができ、陰影光源がパス間でジッタ状にされ、次いで、上記２）で述べたブロック累積バッファを用いてサンプルが平均化される。

【００８０】５）サーチライト効果：これは、シーンの領域に光束を投射するトーチビーム又はヘッドライトの効果である。これは、陰影ボリューム及び光ボリューム

13

の両方を表すことができるように陰影ビットを適当にセットすることにより実施できる。ソフトなエッジは、上記のように模擬することができる。

【0081】テクスチャマップを走査順序で記憶する問題及びそれにより生じるページブレークの問題は、上記で述べた。これについて、以下に、より詳細に説明する。

【0082】所与のシーンピクセルをテクスチャ処理す＊

14

＊るときには、必要なテクスチャピクセル又はテクセル、或いはＭＩＰマッピングの場合には、スクリーンピクセルへとマップされる多数のテクセルを得るために、テクスチャマップをアクセスしなければならない。

【0083】メモリにおける正しいテクセルをアクセスするために、次の式の関数を計算して、テクセルの位置を見つけなければならない。

Ｔｅｘｅｌ＿ａｄｄｒｅｓｓ＝Ｔｅｘｔｕｒｅ＿ｂａｓｅ＿ａｄｄｒｅｓｓ
＋Ｏｆｆｓｅｔ＿Ｆｕｎｃ（ｕ、ｖ）

Ｔｅｘｔｕｒｅ＿ｂａｓｅ＿ａｄｄｒｅｓｓは、テクスチャメモリにおけるＭＩＰ＿ｍａｐレベルの所与のテクスチャのスタート位置であり、そしてｕ、ｖは、計算された整数テクスチャ座標である。Ｏｆｆｓｅｔ＿Ｆｕｎｃは、ｕ、ｖ座標をテクスチャマップのオフセットに対してマップする。

【0084】テクスチャが走査線順序で記憶される場合には、オフセット関数が次のようになる。

Ｏｆｆｓｅｔ＿Ｆｕｎｃ（ｕ、ｖ）＝ｕ＋ｓｉｚｅ＊ｖ

但し、ｓｉｚｅは、ｕ次元のテクスチャマップである。テクスチャマップの次元は２の累乗であるから、上記乗算は、２進表示での単なる左シフト動作であり、そして加算は、ビットの単なる連鎖である。それにより１２８ｘ１２８テクスチャマップに対して得られる関数が図７に走査順序で示されている。（$v_x$は、ｖのｘ番目のビットを指し、最下位ビットは０で番号付けされていることに注意されたい。）

【0085】これに伴う問題は、ｖインデックスが何らかの量だけ変化するや否や、それによりオフセットが急激に変化することである。これらの急激な変化は、非常に多数のページブレークを生じ、これは、テクスチャ処理性能に悪影響を及ぼす。一般に、テクスチャ座標は徐々に変化する。しかしながら、これらはｕ及び／又はｖ方向に同等に変化する傾向がある。

【0086】テクスチャマップを構成する更に効率的な方法は、ｕ及びｖインデックスビットをインターリーブすることである。最下位ビットに$v_0$を有する１つのこのような方法が、最適な関数のもとで図７に示されている。（Ｕでスタートすることもできる。）

【0087】この構成は、多数の隣接ピクセル（ｕ及びｖの両方向に隣接する）をテクスチャメモリにおいて比較的接近して保持し、それ故、ページブレークを減少する。図８は、テクスチャマップの左上の角を示している。各ピクセルの番号は、ピクセルが記憶されるオフセットを示している。

【0088】この構造は、フラクタルとして最も良好に説明される。このパターンは、それ自身で同様に回転された「Ｚ」であり、即ちジグザグがピクセルレベルでスタートし、ピクセルのグループが更に大きくなるのと共に続けられる。

【0089】ＭＩＰマッピングの要求に伴い更に別の問題が生じる。ＭＩＰマッピングのプロセスは、通常、各スクリーンピクセルの計算のたびにＭＩＰマップの２つの隣接レベルをアクセスする必要がある。ＭＩＰマップレベルが隣接して記憶される場合には、２つのレベルからのピクセルフェッチが、１つのＤＲＡＭページ内に留まるにはあまりに離れ過ぎ、従って、各ピクセルアクセスにおいてページブレークが生じる。

【0090】このオーバーヘッドを回避するには、ＤＲＡＭバンクを使用する必要がある。多数のバンクを有するメモリ構成では、一度に２つ以上のページをオープン状態に保持することができる。オープンしたページは異なるバンクになければならないという制約がある。それ故、連続するＭＩＰマップレベルが個別のバンクに記憶される場合には、各ピクセルフェッチのたびにページブレークが生じない。

【0091】以下の説明においては、次のようなネーミング規定を使用する。ＭＩＰマップレベルは０から番号付けされ、０は１ｘ１解像度マップを指し、１は２ｘ２マップを指し、等々となる。図９は、全ての奇数番号のＭＩＰマップレベルを１つのバンクにおける隣接ブロックとして示し、そして全ての偶数レベルを別のバンクにおける隣接ブロックとして示している。

【0092】この構成は、更なるＭＩＰマップテクスチャをメモリに追加するときに問題を生じさせる。バンクＹにおけるメモリ要求は、バンクＸのほぼ４倍である。バンクが同じサイズであるときには、バンクＹは、バンクＸよりかなり前にいっぱいになる。この問題は、奇数ＭＩＰマップレベルが記憶されるバンクをトグルすることにより回避できる。不都合なことに、この構成は、奇数及び偶数のデータブロックに対して個別のベースアドレスを必要とする。

【0093】更に優れた解決策は、図１０に示すように、ＭＩＰマップの対をインターリーブすることである。テクスチャＡは、バンクＸのベースアドレスに最低解像度のマップ（１ｘ１）を有する。次のマップ（２ｘ２）は、バンクＹの連続アドレスにある。テクスチャＢは、バンクＹのベースアドレスに１ｘ１マップを有し、そしてバンクＸの連続アドレスに２ｘ２マップを有する。

15

【００９４】上記のテクスチャマッピング方程式は、テクスチャ処理される各スクリーンピクセルごとに、６つの乗算と、６つの加算と、２つの除算とを必要とする。必要な計算を減少するために、差の式が使用される。

【００９５】入力ピクセル流は、走査順序で送られるので、同じテクスチャ処理関数を共有するピクセルの水平方向スパンがある。Ｘ次元における数１の差の式は、そのスパン内の必要な計算を減少する。

【数１】

$$u(x,y)=\frac{ax+by+c}{px+qy+r}=\frac{\alpha(x,y)}{\beta(x,y)}$$

$$u(x+1,y)=\frac{a(x+1)+by+c}{p(x+1)+qy+r}=\frac{\alpha(x,y)+a}{\beta(x,y)+p}$$

$$v(x,y)=\frac{dx+ey+f}{px+qy+r}=\frac{\lambda(x,y)}{\theta(x,y)}$$

$$v(x+1,y)=\frac{d(x+1)+ey+f}{p(x+1)+qy+r}=\frac{\lambda(x,y)+a}{\theta(x,y)+p}$$

【００９６】スパンのスタートは、完全な計算を必要とするが、その後のピクセルは、３つの加算と、２つの除算しか必要としない。

【００９７】テクスチャマッピング計算に対するアーキテクチャーは、２つのユニット即ち前計算ユニット及び繰り返しユニットに分割される。前計算ユニットは、ａｘ＋ｂｙ＋ｃ、ｄｘ＋ｅｙ＋ｆ及びｐｘ＋ｑｙ＋ｒを評価する。繰り返しユニットは、「ａ」、「ｄ」及び「ｐ」の除算及び加算を実行する。

【００９８】テクスチャの異なる領域における演算精度の目に見える効果を詳細に検査した後に、混合浮動及び固定小数点方法が最も適当であると判断された。

【００９９】係数「ｐ」、「ｑ」及び「ｒ」は整数であり、従って、「ｐｘ＋ｑｙ＋ｒ」の評価は、整数乗算及び整数加算しか必要としない。

【０１００】係数「ａ」、「ｂ」、「ｃ」、「ｄ」、「ｅ」及び「ｆ」は、単一のべき指数２を共有し、これは、他のテクスチャ処理係数と共に記憶される。「ａｘ＋ｂｙ＋ｃ」及び「ｄｘ＋ｅｙ＋ｆ」の評価は、「ｐｘ＋ｑｙ＋ｒ」の評価と同じであるが、べき指数係数も除算ユニットへ送られる。

【０１０１】除算は、浮動小数点で行われる。評価される項「ａｘ＋ｂｙ＋ｃ」、「ｄｘ＋ｅｙ＋ｆ」及び「ｐｘ＋ｑｙ＋ｒ」は、仮数及びべき指数２を伴う浮動小数点数へと換算される。除算の結果は、アドレス発生に用いるための整数へと換算される。

【０１０２】現在の設計は、１６ビットの係数と、１４ビットの逆数の仮数とを使用する。

| オペランド | 逆数 |
|---|---|
| 0x3FFF | 0x2000 |
| 0x3FDF | 0x2010 |

16

【０１０３】２つの除算は、単一の逆数化及び２つの乗算によって行われる。逆数化ユニットへの入力は、０．５ないし０．９９９９の固定小数点数であり、出力は、２ないし１の範囲である。

【０１０４】この関数を解く最も適した方法は、圧縮ルックアップテーブルによるものである。関数は、１４ビットまでの精度であることが必要とされる。これは、０．５が０ｘ２０００（１６進）として表され、０．９９９９が０ｘ３ｆｆｆであることを意味する。データを完全に非圧縮状態で記憶するためには、各々１４ビットの８１９２の位置が必要である（１４ＫＢ）。逆数の最上位ビットは、常に１であるから除去することができる。これは、記憶量を各々１３ビットの８１９２の位置（１３ＫＢ）まで減少する。

【０１０５】データを圧縮する方法は、逆数の差を記憶することである。オペランドが０．７０７より大きい場合には、逆数は１又は０だけ変化する。それ故、これらの差は、１ビットの記憶しか必要としない。０．７０７より小さければ、逆数は、２又は１だけ変化し、これも１ビットで記憶できる。この圧縮方法は、必要なメモリを１．３７５ＫＢまで減少する。以下のテーブルは、幾つかの逆数とそれらの差を示す。

| オペランド | 逆数 | 差 |
|---|---|---|
| 0x3FFF | 0x2000 | - |
| 0x3FFE | 0x2001 | 1 |
| 0x3FFD | 0x2001 | 0 |
| 0x3FFC | 0x2002 | 1 |
| 0x3FFB | 0x2002 | 0 |
| 0x3FFA | 0x2003 | 1 |
| 0x3FF9 | 0x2003 | 0 |
| 0x3FF8 | 0x2004 | 1 |
| 0x3FF7 | 0x2004 | 0 |

【０１０６】データを圧縮解除するために、差の加算が行われる。例えば、０ｘ３ＦＦＢの逆数を計算するために、０ｘ３ＦＦＦないし０ｘ３ＦＦＢに対する差のビットが加算され、次いで、０ｘ２０００（１＋０＋１＋０＋０ｘ２０００≒０ｘ２００２）に加算される。オペランドが０．７０７より下がったときは、差が２又は１となる。これらの差は、単一ビットとして記憶されるが、差を加算するときには２及び１として解釈される。

【０１０７】この圧縮／圧縮解除方法は、非常に多数の加算を必要とするので、少数のクロックサイクルで行うことは不可能である。若干効率の悪い圧縮方法は、加算の数を大巾に減少することができる。３２ごとの逆数値が、介在する３１の数字の差と共にいっぱいに記憶される。以下のテーブルは、このデータの最初の幾つかのエントリーを示す。

| 介在する差 |
|---|
| 10101010101010101010101010101 |
| 10101010101010101010101010101 |

0x3FFF 0x2020 10101010101010101010101010101010

【０１０８】例えば、０ｘ３ＦＤＡの逆数を見つけるために、オペランドは、先ず、それに関連した全逆数をもつ最も近い数（０ｘ３ＦＤＦ）に丸められる。これらの値の差は、加算されるべき差の数（０ｘ２０１０＋１＋０＋１＋０＋１＝０ｘ２０１３）である。

【０１０９】ここに提案する方法は、一連の基本値及びオフセットを記憶する。ｎビットの到来するオペランドは、次のように２つの部分において２つの個別の番号として考えられる。

ａ．番号Ｋと称する最上位ビット、及び

ｂ．番号Ｌと称する最下位ビット。

【０１１０】基本値は、ＫをＲＯＭアドレスとして用いてＲＯＭから読み取られ、そしてオフセットは、同じアドレスにおいて別のＲＯＭ（「デルタＲＯＭ」と称する）から読み取られる。

【０１１１】次いで、この方法は、デルタＲＯＭからのデータの最初のＬビットに生じる１の数をカウントし、これは、この基本値からのオフセットを与え、逆数に到達するためにはこのオフセットを基本値に加えなければならない。オフセット値が上記ケース１に対応する場合には、次のようになる。

（ｉ）オフセット＝（デルタＲＯＭの最初のＬビットにおける１の数の和）。

上記のケース２に対応する場合には、次のようになる。

（ii）オフセット＝（デルタＲＯＭの最初のＬビットにおける１の数の和＋Ｌ）。この第２のケースは、１又は２を表すデルタＲＯＭからの値を効果的に補正するが、それらは、２進で「０」又は「１」として表される。

【０１１２】ハードウェアの構成が図１３に示されている。ｐｘ＋ｑｙ＋ｒの値（「ｐｑｒ積」として示された）は、１．０ないし２．０の値に正規化される（４０）。仮数は、上記のように、２つの部分、即ち最上位部分及び最下位部分において考えられる。最上位部分は、ＲＯＭ４２、４４をアクセスするアドレスとして使＊用される。

【０１１３】最下位部分は、マスク４６を発生するのに使用され、これは、デルタＲＯＭ４４からの値と論理的にアンドされる（４８）。これは、デルタＲＯＭから最初のＬビットを分離し、それらをカウントできるようにする（５０）。ユニットから出て来るカウントされた１の数は、５２Ｌをこれに加算することにより計算される。次いで、$\sqrt{2}$の元の数を比較する比較器によって正しいオフセットが選択され、式（ｉ）又は（ii）から適切な結果が選択される。

【０１１４】このオフセット値は、次いで、「メインＲＯＭ」からの基本値に加えられ（５６）、「ｐｑｒ積」の逆数が与えられる。

【０１１５】この図は、ａｘ＋ｂｙ＋ｃ（「ａｂｃ積」）によって与えられる逆数を乗算し（５８）そして正規化解除する（６０）ことにより、次の数２の式をいかに行うかを示している。

【数２】

$$u = \frac{ax+by+c}{px+qy+r}$$

【０１１６】Ｖは、ｄｘ＋ｅｙ＋ｆ（「ｄｅｆ積」）を用いて同様に計算される。

【０１１７】「Ｄ」を計算する計算経費を回避するために、精度は低いが効率の良い方法は次の数３である。

【数３】

$$D = \frac{n}{(px+qy+r)^2}$$

「ｐｚ＋ｑｙ＋ｒ」はテクスチャマッピング方程式の下半分である。「ｎ」は、レンダリングの前に評価されて他のテクスチャ処理パラメータと共に記憶される定数である。「ｎ」を計算する方法は多数あり、例えば、次の数４がある。

【数４】

$$n = r^2 \, MAX \, \frac{ar-pc^2}{pr+r^2} + \frac{dr-pf^2}{qr+r^2} \, , \quad \frac{br-qc^2}{qr+r^2} + \frac{er-qf^2}{qr+r^2}$$

「ａ」、「ｂ」、「ｃ」、「ｄ」、「ｅ」、「ｆ」、「ｐ」、「ｑ」及び「ｒ」は、テクスチャマッピング係数である。

【０１１８】上記の数４は、ピクセル座標（０、０）においてヘクベルトの近似を使用して「Ｄ」を評価し、次いで、数３の式をその値に等しくする。これは理想的なものではない。というのは、「Ｄ」の最良の近似は、テクスチャ処理面が見えるところでなければならないからである。それ故、良好な方法は、数３の式を、見える面のピクセル座標においてヘックバートの近似に等しくすることである。

【０１１９】図１１に示した累乗関数近似のハードウェア実施について説明する。これは、固定小数点表示で表された０≦Ｘ≦２の範囲の値Ｘと、小さな正の係数Ｋを入力として受け取り、そして固定小数点の結果を再び０≦Ｘ≦２の範囲で出力する。

【０１２０】ビットＸの数（即ち、図の点Ａにおける関数への入力）は、ハイライトの所要濃度によって左右さ

19

れるが、通常は、典型的に、約１６ビットである。この説明においては、これがmである。

【０１２１】ｋのビット数は、通常、３又は４であり、これも、ハイライトの所要濃度によって左右される。これは、ｎと称する。

【０１２２】結果のビット数は、所要の精度に基づいている。ｐビットの出力精度が必要とされると仮定する。

【０１２３】図１１のプロセスは、次の通りである。

ステップ

１．固定小数点ｘが、最初に、１の固定小数点定数値から減算される。出力のビット数は、入力と同じに保たれ、即ちmビットである。

２．次いで、値は、ｋ位置だけ左へシフトされる。点Ｂの巾は、ｍ＋（２π－１）ビットであって、０．．．２πの範囲の固定小数点数を表している。

３．次いで、値は、最大関数により１にクランプされる。これは、上位の２πビットを検査し、いずれかがセットされている場合には、その値を１以上にしなければならない。この例では、出力を固定点１．０にセットすることができる。上位の２πビットがどれもセットされていない場合には、出力値は入力と同じである。必要とされる結果の質に基づいて、点Ｃにおいて最大関数から出力されるビットの数は、ｐ又はｐ＋１ビットに限定される。選択されるビットは、下位mの上位ｐ又はｐ＋１ビットである。即ち、値の出力は、依然として、０≦ｘ≦１の範囲の固定小数点数であるが、精度はｐ又はｐ＋１に過ぎない。

４．必要とされる結果の質に基づいて、点Ｃにおいて最大関数から出力されるビットの数は、ｐ又はｐ＋１ビットに限定される。選択されるビットは、下位mの上位ｐ又はｐ＋１ビットである。即ち、値の出力は、依然として、０≦ｘ≦１の範囲の固定小数点数であるが、精度はｐ又はｐ＋１に過ぎない。

５．次いで、値は、固定小数点フォーマットにおいて１＊から減算される。精度のビット数は、依然として、ｐ又はｐ＋１に保たれる。

20

６．次いで、値は、それ自身を乗算することにより平方される。オペランドのサイズ、即ちｐ又はｐ＋１ビットに基づいて、これは、$p^2$ 又は $(p+1)^2$ ビットをもつ値を与える。いずれにせよ、上位のｐビットが取り出され、０≦結果≦１の範囲の固定小数点の値を与える結果として出力される。

【０１２４】陰影付けハードウェアユニットは、標準的な差動技術である前計算及び繰り返しユニットに対してはテクスチャ処理ユニットと相違しない。フラット、一次関数、二次関数の２つの陰影付けモードについて以下に説明する。

【０１２５】以下の説明上、陰影付け関数をｘ及びｙのピクセル位置について説明する。これらは、絶対的なスクリーンピクセル位置であるが、陰影付けされる面の中心に対する位置にするのが更に効果的である。このようにすると、以下に述べる種々の$T_n$パラメータを表すのに必要な精度が相当に低減される。

【０１２６】フラットな陰影付けの場合には、前計算も繰り返しも行う必要がなく、定数値が直接使用されるか、又はテクスチャ処理パイプラインの結果により乗算されるかのいずれかである。

【０１２７】一次関数陰影付けは、次の式の前計算を必要とする。

$$I(x,y)=T_2\,x+T_1\,y+T_0$$

【０１２８】従って、繰り返し部分は、$T_2$ を繰り返し加算するだけでよい。というのは、次の通りだからである。

$$\begin{aligned}I(x+1,y)&=T_2\,(x+1)+T_1\,y+T_0\\&=I(x,y)+T_2\end{aligned}$$

【０１２９】二次関数陰影付けも、差の方程式を用いて行われる。二次関数陰影付けの式は次の通りである。

$$I(x,y)=T_5\,x^2+T_4\,xy+T_3\,y^2+T_2\,x+T_1\,y+T_0$$

従って、ピクセル（ｘ、ｙ）とピクセル（ｘ＋１、ｙ）※ ※との間のＩの差は、次の式で与えられる。

$$\begin{aligned}\Delta I(x,y)&=I(x+1,y)-(x,y)\\&=T_2\,(x+1)^2-T_2\,x^2+T_4\,(x+1)\,y-T_4\,xy\\&\quad+T_2\,(x+1)-T_2\,x\\&=T_2\,(2x+1)+T_4\,y+T_2\end{aligned}$$

それ故、ピクセルｘｙにおける差の間の差は、次の通り★ ★である。

$$\begin{aligned}\Delta\Delta I(x,y)&=\Delta I(x+1,y)-\Delta I(x,y)\\&=T_5\,(2(x+1)+1)+T_4\,y+T_2\\&\qquad-T_5\,(2x+1)-T_4\,y-T_2\\&=T_5\,(2x+3)-T_5\,(2x+1)\\&=2T_5\end{aligned}$$

従って、前処理ユニットは、Ｉで始まる初期値と、初期☆ ☆デルタ値を計算することが必要である。即ち、

$$I(x,y)=T_5\,x^2+T_4\,xy+T_3\,y^2+T_2\,x+T_1\,y+T_0$$

$$\Delta I(x,y)=T_5\,(2x+1)+T_4\,y+T_2$$

繰り返しを行うために、デルタ値を加算することにより 次のＩ値が得られ、デルタの差において加算することに

より新たなデルタが発生される。即ち、

$$I(x+1, y) = I(x, y) + \Delta I(x, y)$$
$$\Delta I(x+1, y) = \Delta I(x, y) + \Delta\Delta I(x, y)$$
$$= \Delta I(x, y) + 2T_5$$

ピクセルごとに２回の加算を行うだけでよい。出力値は固定小数点であり、範囲０．．．１にクランプされる。ハイライトを計算する場合には、この値を上記のように累乗関数に入れることができる。

【０１３０】上記したように、霧付与の減衰係数は、次の式によって計算される。

$A = d^n$

但し、Ａは減衰係数であり、ｄは霧の濃度であり、そしてｎはレンダリングされるオブジェクトと観察者との間の距離である。この式は、次のように書き直すことができる。

$A = 2^{n \cdot \log_2(d)}$

これは、乗算及び$2^{-x}$関数に対する演算を簡単化する。この関数をルックアップテーブルで具現化する場合には、必要とされる精度に対して甚だしく大規模なものとなり、従って、情報の圧縮方法が提案される。

【０１３１】関数$2^{-x}$は、便利なことに、２の累乗ごとに「自己類似」である。０と１との間の$2^{-x}$は，１と２との間の$2^{-x} * 2$と同じであり、且つ２と３との間の$2^{-x} * 4$と同じである。それ故、この関数は、２の分数累乗のルックアップテーブルと出力に対する演算シフト回路とによって具現化することができる。提案された関数への入力の一例を以下に示す。

ビット０　１　２　３　４　５　６　７　８　９　１０　１１　１２　１３

１４　１５　１６　１７　１８　１９　２０　２１　２２　２３　２４　２５・・・・

ビット０－６は、累乗テーブルへのインデックスである。ビット７－９は、ルックアップの答えをシフトダウンするのに使用される。１０より上のビットがセットされた場合には、結果が０である。

【０１３２】上記特許出願「三次元像の陰影付け」（出願第９４１４８３４．３号）に開示されたアーキテクチャーは、レンダリングされるオブジェクトと観察者との間の距離の逆数に比例する数字を与えることができる。霧付与関数は、$2^{\log_2(d)/n}$に変更しなければならない。不都合なことに、この関数を容易に評価する関数はない。この関数に対する受け入れられる近似は、$A = 1 - 2^{n \cdot \log_2(d)}$である。

【図面の簡単な説明】

【図１】ＭＩＰマップデータの種々の解像度を示す図である。

【図２】拡散陰影付けの図である。

【図３】鏡状ハイライトの図である。

【図４】陰影付けに使用されるシステムアーキテクチャーのブロック図である。

【図５】メモリアドレスの従来の走査と、アドレスビットをインターリーブした走査とを比較するための図である。

【図６】メモリアドレスの従来の走査と、アドレスビットをインターリーブした走査とを比較するための図である。

【図７】ビットが実際にいかにインターリーブされるかを示す図である。

【図８】物理的なメモリ位置がインターリーブされたアドレスビットでアドレスされる順序を詳細に示す図である。

【図９】２つのメモリバンク内の１組のＭＩＰマップデータの記憶を示す図である。

【図１０】２つのメモリバンク内の２組のＭＩＰマップデータの記憶を示す図である。

【図１１】累乗霧付与関数を具現化するブロック図である。

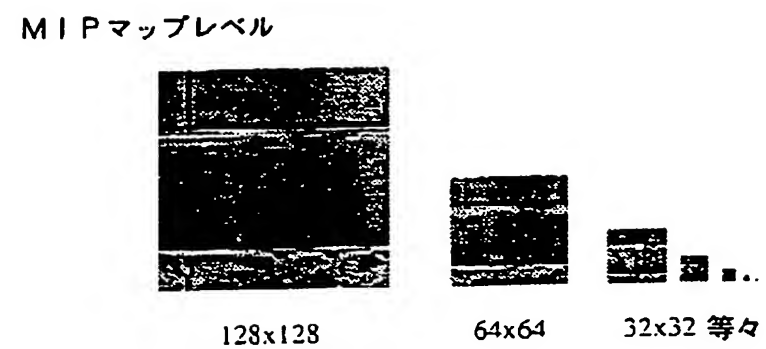
【図１２】図４の詳細な実施を示す図である。

【図１３】除算演算を実行する改良された回路のブロック図である。

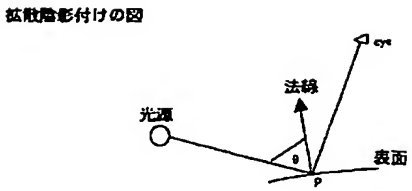
【符号の説明】

４　パラメータキャッシュ
６　テクスチャメモリ
８　前計算ユニット
１０　テクスチャ繰り返しユニット
１２　陰影付けユニット
２０　入力データ
２２　タグバッファ
２６　パラメータキャッシュ
２８、３０　テクスチャ／陰影付け繰り返し装置
３２　前計算バッファ
３４　ピクセル処理ユニット
３６　霧付与ユニット
３８　ブロック累積バッファ

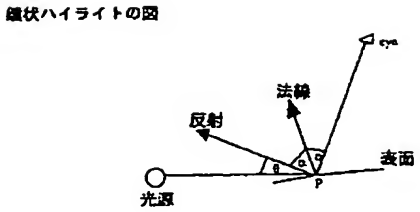
【図１】



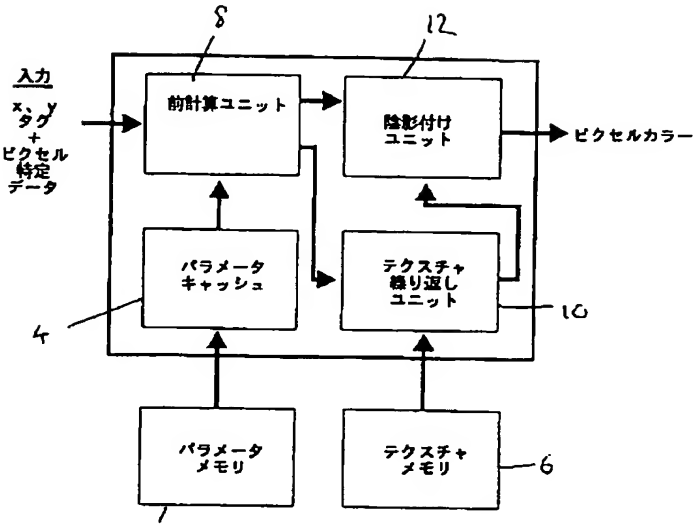

【図２】




【図３】

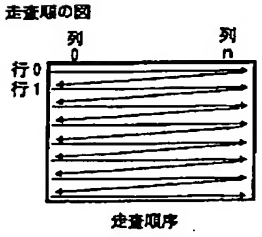



【図４】

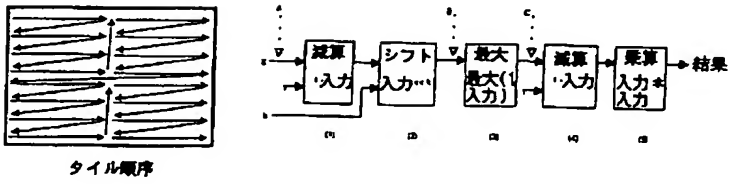



【図５】

【図６】

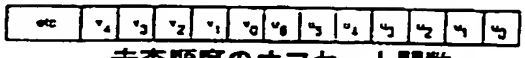



【図１１】

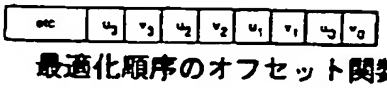



【図７】

テクスチャインデックスビット配列



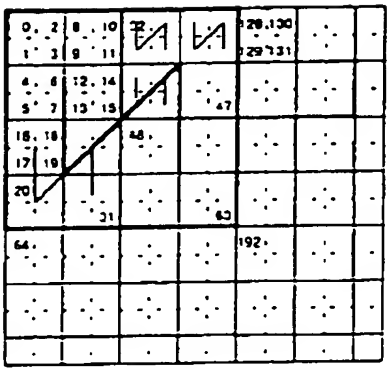

走査順序のオフセット関数



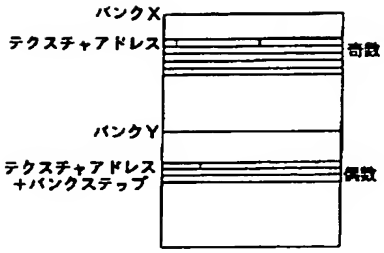

最適化順序のオフセット関数

【図８】

メモリのＭＩＰマップレベル配列

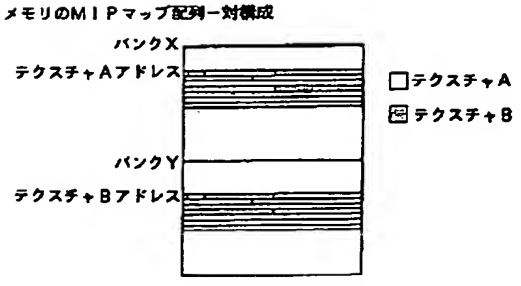


【図９】

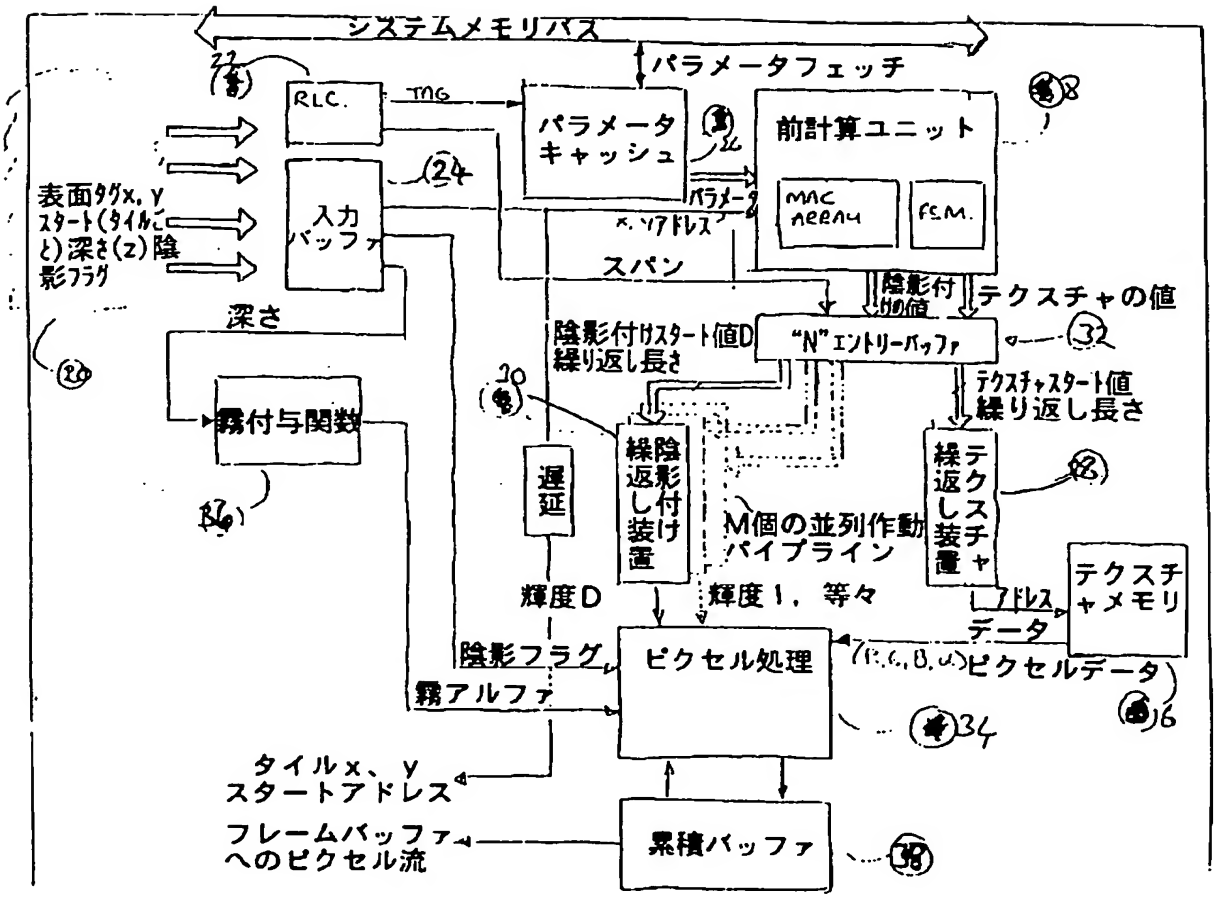



メモリのＭＩＰマップ配列－パック状態

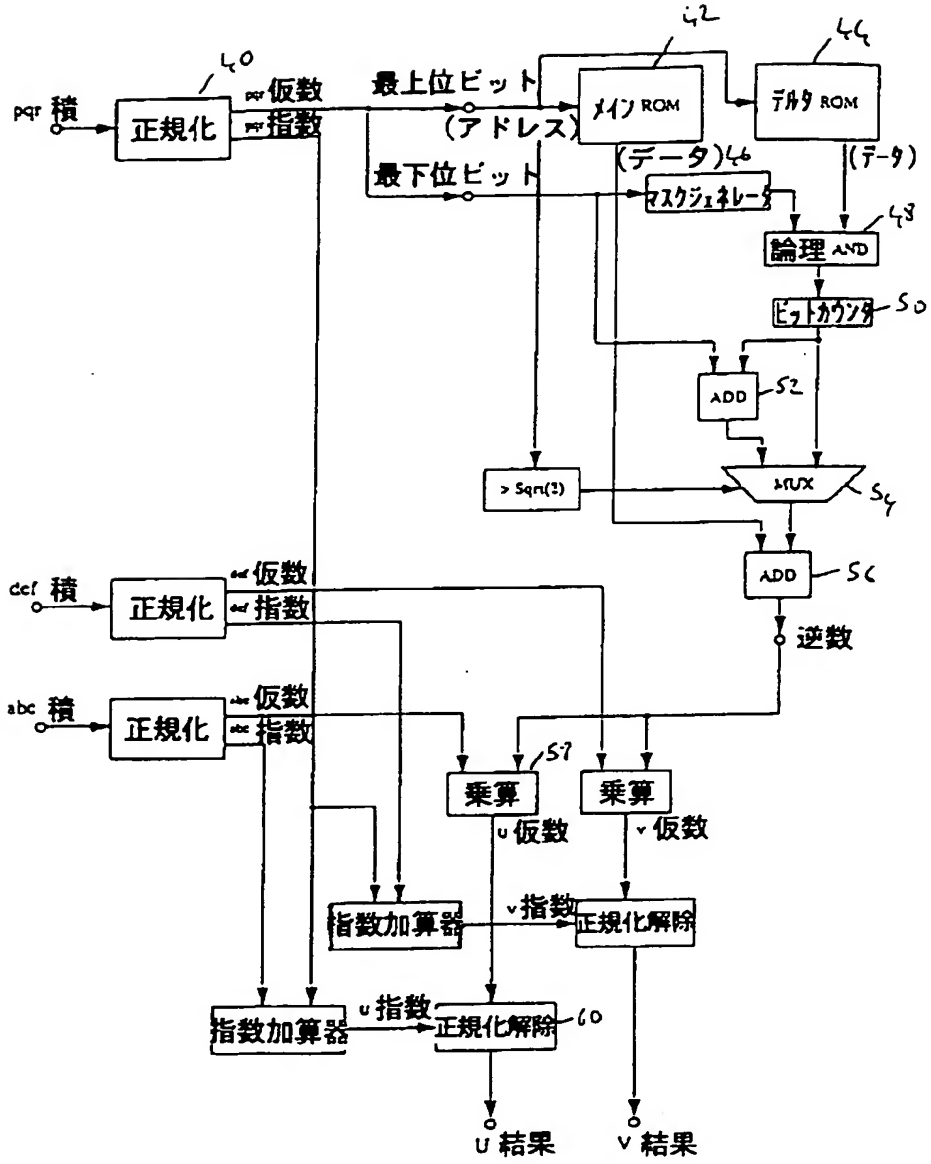
【図１０】



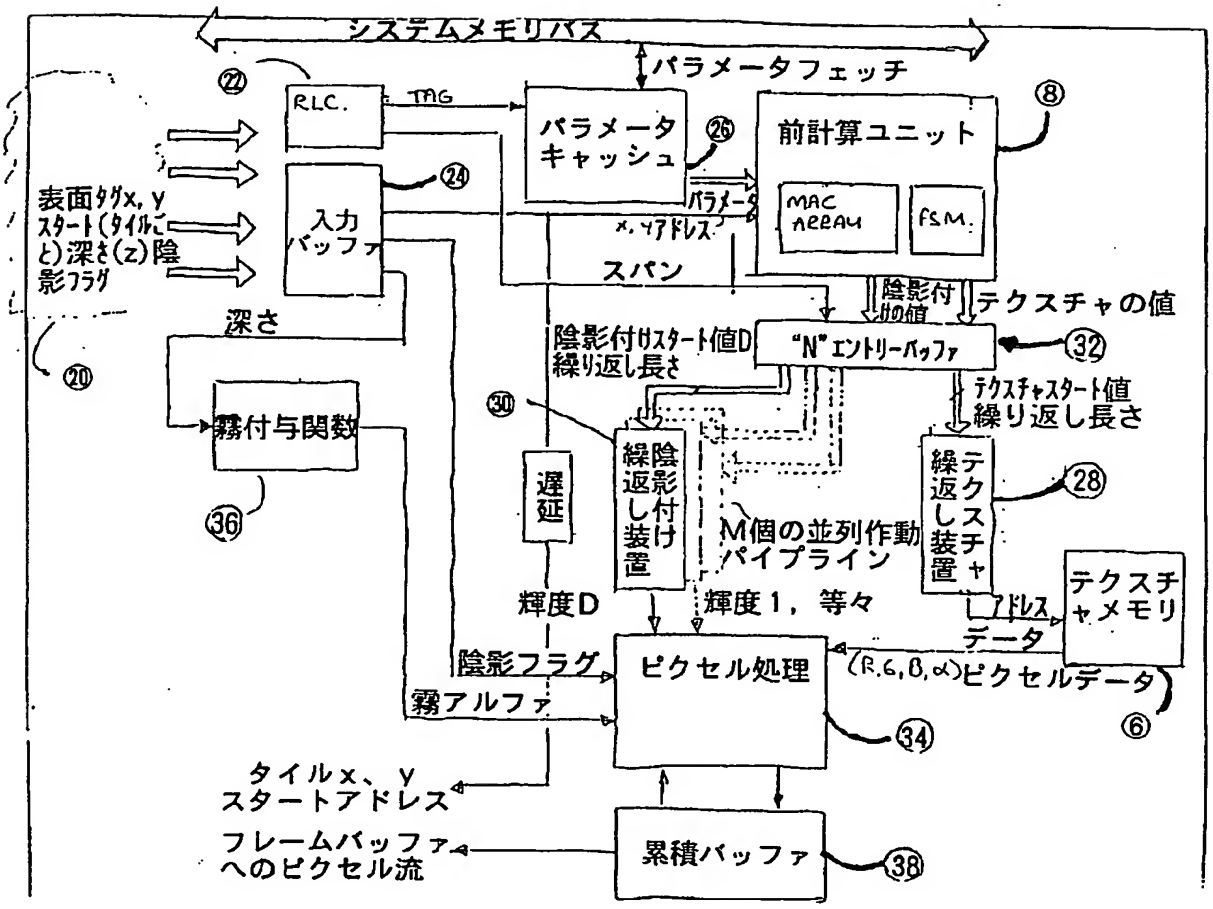
【図１２】


システムメモリバス
パラメータフェッチ
RLC.
TMG
パラメータキャッシュ
前計算ユニット
MAC AREA
FS.M.
表面タグx, y スタート(タイルごと)深さ(z)陰影フラグ
入力バッファ
パラメータ
メモリアドレス
スパン
陰影付け値
テクスチャの値
深さ
陰影付けスタート値D
繰り返し長さ
"N"エントリーバッファ
霧付与関数
遅延
繰返し陰影付け装置
M個の並列作動パイプライン
テクスチャスタート値
繰り返し長さ
繰返しテクスチャ装置
テクスチャメモリ
輝度D
輝度I，等々
アドレス
データ
陰影フラグ
ピクセル処理
(R,G,B,α)ピクセルデータ
霧アルファ
タイルx、y スタートアドレス
フレームバッファへのピクセル流
累積バッファ

【図１３】




【手続補正書】
【提出日】平成８年４月４日
【手続補正３】
【補正対象書類名】図面
【補正対象項目名】図１２
【補正方法】変更
【補正内容】
【図１２】




---

フロントページの続き

(72)発明者　マーク　エドワード　ダン
イギリス　ハートフォードシャー　ダブリューディー1　3ティーエヌ　ワトフォード　ザ　リッジウェイ　26

(72)発明者　イアン　ジェームズ　オーヴァーリーズ
イギリス　ロンドン　ダブリュー9　3ディーピークイーンズ　パーク　アッシュモア　ロード　211

(72)発明者　ピーター　ディヴィッド　リーパック
イギリス　ハートフォードシャー　ダブリューディー7　3ティーエヌ　ラドレット　リンクス　ドライヴ　5

(72)発明者　ホッセイン　ヤーセイ
イギリス　バッキンガムシャー　エイチピー5　2アールエス　チェーシャム　パークレイ　アベニュー　111